Unleashing the Potential of Deep Learning for Accurate Brain Tumor Classification: A Comprehensive Study and Model Development

Jesty Sebastian
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
jesty.sebastian@stu.mmu.uk

***Abstract*— The classification of brain MRI scans into the four groups of glioma, meningioma, no tumor, and pituitary is the main goal of this study. A deep learning model based on convolutional neural networks (CNNs) was created for precise categorization using a dataset of 7023 photos downloaded from https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset. Numerous tests were run to assess the model's performance using different measures. The outcomes show that the suggested model outperformed current methods with great accuracy. This study advances deep learning methods for classifying brain tumors, perhaps assisting with accurate diagnosis and treatment choices. Expanding the dataset and looking into interpretability techniques for clinical applications could be part of further investigation.**

Keywords-Deep Learning, Tumor, Glioma, Meningioma, No-Tumor, Pituitary, MRI, CNN, VGG, ResNet, AlexNet

I. Introduction

Brain tumors are a significant medical challenge, affecting millions of individuals worldwide and posing a considerable burden on healthcare systems. The diagnosis, planning of the course of treatment, and patient monitoring all depend on the accurate classification of images of brain tumors. Traditional approaches for brain tumor analysis frequently rely on manual interpretation by radiologists, which can be time-consuming, subjective, and subject to error.

Deep learning, a branch of artificial intelligence, has made recent breakthroughs that have showed promise for enhancing the precision and effectiveness of tumor analysis. Convolutional neural networks (CNNs), in particular, have shown amazing ability to extract relevant features from medical imaging data, resulting in improved classification performance.

This work uses a dataset of 7023 MRI scans of the human brain to investigate the use of deep learning algorithms for the classification of brain tumor pictures. The dataset has four unique classes: pituitary, meningioma, glioma, and no tumor. We are working to create a model that is capable of correctly recognizing and categorizing brain tumors by utilizing the power of deep learning.

The main goals of this study are to examine how well deep learning models perform at identifying images of brain tumors, compare their precision to that of current methods, and estimate how well they might revolutionize the detection and treatment of brain tumors. We seek to build and optimize a deep learning model that can give accurate and effective classification of brain tumor images by utilizing the enormous quantity of labelled data present in the dataset.

This research also intends to add to the body of knowledge in the area of deep learning for medical image interpretation, specifically in the context of brain tumors. As an accurate and effective classification system can help clinicians make educated decisions and enhance patient outcomes, the findings of this study may have significant implications for the medical community.

This paper discusses the dataset and methods used, including the architecture and training process of the deep learning model, and gives an overview of related studies and developments in deep learning for brain tumor analysis. It examines the implications and prospective applications of the constructed model in addition to presenting the experimental findings and performance evaluation. The article concludes by summarizing the findings of this study and outlining possible future routes for improvements in the field of deep learning-based brain tumor classification.

II. DATASET DESCRIPTION AND ANALYSIS

The dataset utilized in this study is a comprehensive compilation that incorporates the SARTAJ dataset, Br35H, and figshare. It comprises a total of 7023 MRI scans of the human brain, categorized into four distinct classes: glioma, meningioma, no tumor, and pituitary. Notably, the images belonging to the "no tumor" class were exclusively sourced from the Br35H dataset. The dataset is organized into two folders, namely Training and Testing, with both folders encompassing the same four classes of glioma, meningioma, no tumor, and pituitary. This dataset provides a rich resource for training and evaluating brain tumor classification models, facilitating research and advancements in the field of medical image analysis.

This dataset of brain MRI images provides researchers with a wide range of data, offering a valuable resource for investigating and developing reliable algorithms in the field of brain tumor classification. The dataset's size and diversity play a crucial role in enabling researchers to explore and analyze various types of brain tumors, their characteristics, and their manifestations in medical images.

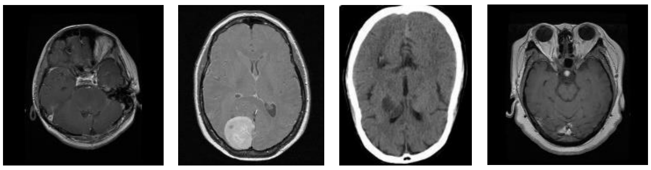

Figure 1 - Illustration of different MRI Scans. From left to right: Glioma, Meningioma, No Tumor, Pituitary

By leveraging this comprehensive dataset, researchers can extract meaningful features, develop advanced machine learning models, and evaluate different classification techniques to accurately identify and differentiate different types of brain tumors. The availability of such a sizable and varied dataset is essential for advancing the field of brain tumor analysis and can significantly contribute to the development of more efficient diagnostic and therapeutic approaches, leading to improved patient care and outcomes.

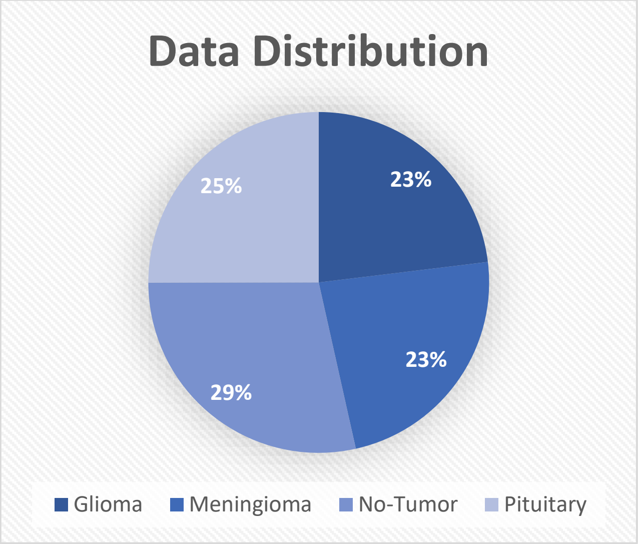

Figure 2 - Distribution of Images in Different Classes

A. Background

Brain tumor classification involves the utilization of traditional machine learning methods, image processing techniques, and deep learning approaches. Traditional machine learning methods[2] such as SVM, Decision Trees, Random Forests, and k-NN offer interpretability and generalizability but may have limitations in terms of computational complexity and overfitting. Image processing techniques[5], including edge detection, region-based segmentation, and thresholding, extract relevant features from medical images, enabling precise tumor boundary delineation and quantitative analysis. Deep learning techniques, specifically CNNs, have shown remarkable effectiveness in automatically learning hierarchical representations from brain tumor images. Deep learning architectures like AlexNet, VGGNet, ResNet, and U-Net have been developed for brain tumor classification. Hybrid approaches, combining traditional machine learning and deep learning methods[8], have also been explored to enhance classification performance. These approaches aim to leverage the strengths of both approaches, resulting in improved accuracy and effectiveness in brain tumor classification tasks.

In this project, Convolutional Neural Networks (CNNs) are chosen for brain tumor classification due to their ability to automatically extract relevant features from medical images. CNNs are particularly well-suited for image data classification tasks as they can effectively capture spatial relationships and patterns within the images. Unlike traditional machine learning algorithms, which require manual feature extraction, CNNs can learn to extract features directly from the raw input data[4], eliminating the need for explicit feature engineering. This makes CNNs highly suitable for medical imaging tasks where the relevant features may be complex and difficult to define explicitly.

Furthermore, CNNs have demonstrated superior performance in various image classification tasks, including medical image analysis. Their hierarchical architecture[13], consisting of multiple convolutional and pooling layers, enables them to learn increasingly complex and abstract representations of the input images. This capability is beneficial for distinguishing between different tumor types and accurately classifying brain tumor images.

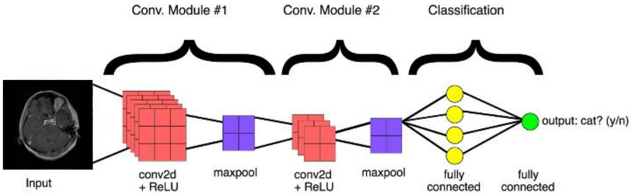

Figure 3 - CNN Architecture

Moreover, CNNs can handle the challenges associated with limited dataset sizes and variability in tumor appearances. By training on a large dataset, CNNs can learn generalized representations and overcome overfitting issues. Additionally, their ability to capture spatial information and local features allows them to adapt to variations in tumor size, shape, and intensity.

Although deep learning models like CNNs can be seen as "black boxes" due to their complexity[7], their performance and ability to automatically learn relevant features outweigh this limitation. The focus is often on the model's accuracy and generalization capabilities, which are crucial in clinical applications. Overall, CNN models are selected for this project because of their capacity to handle complex image data, automatically extract features, and achieve high classification accuracy in brain tumor analysis.

III. Methodology

The experiment was run on VGG11, VGG16, VGG19, AlexNet, and ResNet, and the ones that produced the best results were explained. The models are the VGG11, AlexNet, and, ResNet models.

A. Preprocessing Steps

Several preprocessing processes were used to improve the quality and standardize the input data before the photos were fed into the deep learning network. The image is normalized by being mapped to a range between 0 and 1, achieved by finding the minimum and maximum pixel values in the image and then rescaling the pixel values accordingly. By applying this normalization technique, the image data is transformed to have a consistent and standardized representation, which can aid in the training and convergence of machine learning models. The normalization of the image helps mitigate the impact of varying pixel value ranges and ensures that the model can effectively learn from the data.

B. Data Augmentation Techniques

Using PyTorch's transforms module, two sets of transformations—transform_train and transform_test—are defined. The images are resized to a standard size of 32x32 pixels, transformed into tensors, and their pixel values are normalised using mean and standard deviation values that are unique to the dataset as part of the transform_train transformations. These adjustments guarantee consistency and assist in stabilizing the training procedure. The test images are processed and assessed uniformly using transform_test transformations, which are used in a similar way. These transformations are used to preprocess the images into a deep learning model-friendly format, enabling efficient training and precise predictions.

C. VGG Model

A custom Visual Geometry Group (VGG) model is established. The VGG class has a feature extraction component and a classifier component[6], and it derives from nn.Module. The convolutional layers of the VGG model are represented by the features attribute and are determined by the configuration given. The adaptive average pooling process used by the avgpool layer results in a fixed-size feature map.

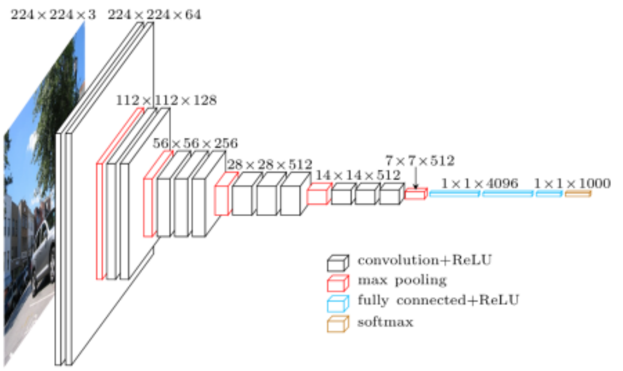

Figure 4 - VGG Architecture

The classifier consists of a series of fully connected layers with dropout regularisation and ReLU activation. The model's forward pass, during which the input is processed by the feature extraction, average pooling, and classifier layers, is defined by the forward method.

The get_vgg_layers function creates the convolutional layers based on the configuration supplied and whether batch normalisation is employed. Each time the configuration list is iterated over, the layers list is expanded to include either a max pooling layer or a convolutional layer followed by batch normalisation and ReLU activation.

1) Rationale Behind VGG Model Design

The rationale behind the VGG model's design is to increase depth while utilising relatively small receptive fields (3x3 convolutional filters) to capture fine-grained features in images[12].

The following are the main driving forces behind the model's development:

Depth for Feature Extraction: By stacking multiple convolutional layers, the VGG model architecture aims to increase the depth of the network. This allows the model to learn hierarchical representations of images at various levels of abstraction. A deeper network enables the model to capture more complex and abstract features, improving performance on image classification tasks.

Small Receptive Fields: The use of small 3x3 receptive fields in the convolutional layers helps in the capture of local patterns and details in images. The network can learn more expressive features by nonlinearly composing them through stacking several 3x3 filters. This design decision enables the model to efficiently learn both low-level features (such as edges, textures, and object parts) and high-level features (such as structures, object parts), in a hierarchical manner.

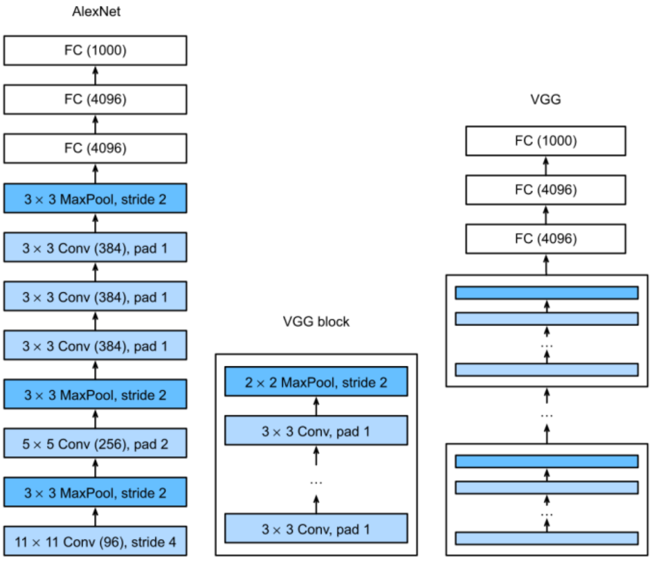

Figure 5 - VGG Architecture

Max Pooling for Spatial Aggregation: The VGG model adds max pooling layers after every two convolutional layers throughout the network. Max pooling helps in spatial downsampling, reducing the spatial dimensions of the feature maps while preserving the most salient features. As a result, the model can achieve some spatial invariance and become more robust to changes in the positions and sizes of objects in the input images.

Fully Connected Layers for Classification: In the classifier portion of the model, the feature maps obtained from the convolutional layers are flattened and run through a series of fully connected layers. In order to classify data, these fully connected layers learn discriminative representations and map them to the desired classes. In the fully connected layers, the use of ReLU activation and dropout regularisation promotes non-linearity and prevents overfitting, respectively.

2) VGG16 Model

A model with batch normalization is built on top of the VGG16 architecture[11]. It defines the convolutional layers of the VGG16 network using the get_vgg_layers function and then assigns them to the vgg16_layers variable. The model is then built using these layers and the four output classes. In addition, the input features of a pretrained VGG16 model with batch normalization's final fully connected layer are loaded. A new fully connected layer is created to customize the pretrained model for the specific classification task.

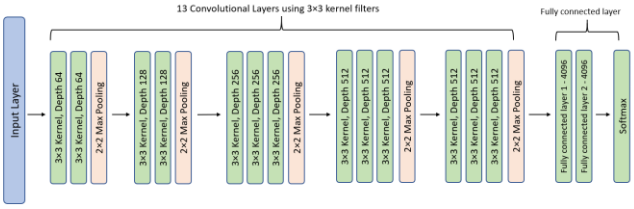

Figure 6 - VGG11 Architecture

a) Training Procedure and Optimization Techniques

In this model, the CrossEntropyLoss criterion is used to calculate the loss. Model is trained using the Adam optimizer with a learning rate of The learning rate is set to 5e-4, and the model parameters are grouped into two sets: the features (convolutional layers) and the classifier (fully connected layers). The learning rate for the features parameters is one-tenth of the overall learning rate, while the learning rate for the classifier parameters remains unchanged. This setup allows for differential learning rates, enabling fine-tuning of the model.

The training process consists of 10 epochs. In each epoch, the model is trained on the training dataset using the train() function, which calculates the training loss and accuracy. Then, the model is evaluated on the validation dataset using the evaluate() function to compute the validation loss and accuracy. The training and validation losses are recorded in separate lists for monitoring the model's performance.

If the current validation loss is lower than the best recorded validation loss so far, the model's state dictionary is saved as 'best-model.pt'. This allows the best-performing model to be saved for future use.

The duration of each epoch is measured using the epoch_time() function, and the training and validation metrics (loss and accuracy) are printed for each epoch. The primary goal of the training procedure is to minimize the training loss while improving the model's accuracy on both the training and validation datasets.

3) *VGG19 Model*

The model is constructed by initializing the VGG19 layers with batch normalization. It has an output dimension of 4, representing the number of classes for tumor classification. A pretrained VGG19 model[9] with batch normalization is loaded, and the input features for the final fully connected layer are determined from the last layer of the pretrained model's classifier. The model design leverages the powerful feature extraction capabilities of VGG19 and adds a custom fully connected layer for the specific classification task. This design allows the model to learn intricate features from input images and make accurate predictions for brain tumor classification.

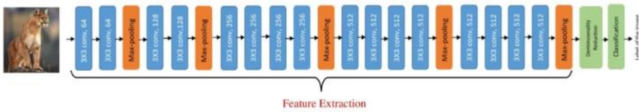

Figure 7 - VGG19 Architecture

a) Training Procedure and Optimization Techniques

In the training procedure, the model is trained using the Adam optimizer with a learning rate of 5e-4. The model parameters are divided into two sets: the features (convolutional layers) and the classifier (fully connected layers). The learning rate for the features is set to one-tenth of the overall learning rate, while the learning rate for the classifier remains unchanged. This allows for differential learning rates between the two sets of parameters.

The training process is executed for 10 epochs. During each epoch, the model is trained on the training dataset, and the train() function is used to compute the training loss and accuracy. The model is then evaluated on the validation dataset using the evaluate() function to calculate the validation loss and accuracy. The training and validation losses are stored in separate lists for tracking the model's performance.

If the current validation loss is lower than the previously recorded best validation loss, the model's state dictionary is saved as 'best-model.pt'. This ensures that the best performing model is retained for future use.

The epoch time is measured, and the training and validation metrics (loss and accuracy) are printed for each epoch. The training procedure aims to minimize the training loss and improve the model's accuracy on both the training and validation datasets.

D. ResNet Model

ResNet is a deep convolutional neural network architecture[3] that was first introduced in 2015 and is renowned for its efficiency in deep model training. For learning residual mappings and enhancing gradient flow, it introduces skip connections.

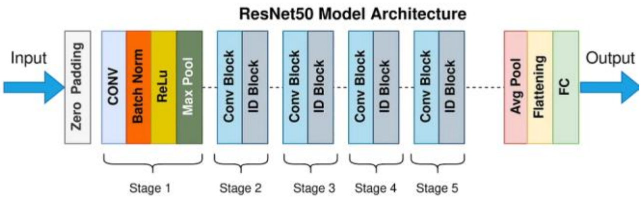

Figure 8 - ResNet50 Architecture

The residual block with shortcut connections, which is depicted in the image below, is the fundamental building block of a ResNet architecture[10].

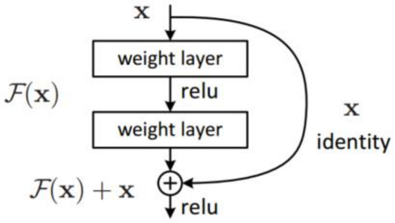
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 9 - Residual Block of ResNet

The model begins with convolutional layers, stacks these building blocks and, concludes with global average pooling and a fully connected layer. Due to its deep structure and skip connections, ResNet has excelled in computer vision tasks.

The ResNet-50 architecture underpins this model. It replaces the pre-trained ResNet-50 model's final fully connected layer with a new fully connected layer with the same number of output features as the number of classes in the dataset. The model can now be fine-tuned for a specific classification task thanks to this modification.

The ResNet model is built with an initial convolutional layer, followed by batch normalisation, ReLU activation, and max pooling. Then it has four layers, each with a different number of residual blocks, as specified in the ResNetConfig. The residual blocks are made up of convolutional layers, batch normalisation, and skip connections, which allow the model to learn residual functions.

The AdaptiveAvgPool2d layer is used to perform adaptive average pooling, which ensures that the model can handle input images of varying sizes. Finally, the output is passed through the fully connected layer to obtain the predicted class probabilities.

The pre-trained ResNet-50 model is used in this model design, which also initializes the fully connected layer and loads the pre-trained weights. The network can benefit from the features learned from a large dataset by using a pre-trained model, which aids in better feature extraction and classification performance on a specific task.

a) Rationale Behind Model Design

The ResNet (Residual Network) model was created to address the issue of vanishing gradients and the degradation of accuracy as network depth increases. The key concept is to use skip connections, also known as "shortcut connections," to allow information to flow directly from earlier layers to later layers. These skip connections create shortcut paths for the gradient during backpropagation, making deep network training easier.

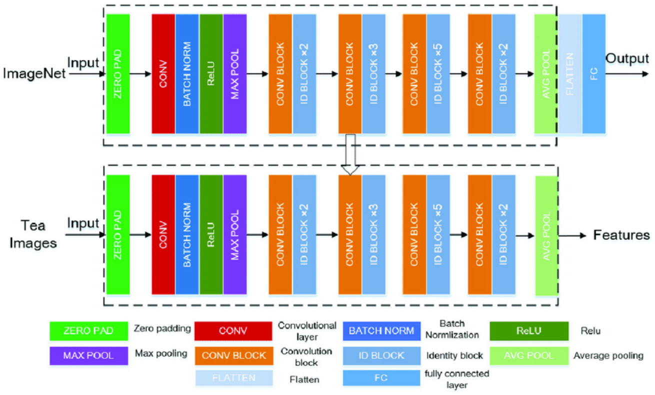

Figure 10 - ResNet Architecture

ResNet models enable the network to learn residual functions rather than directly fitting the desired underlying mapping by introducing residual blocks, which consist of identity mappings and a residual mapping. This method allows the model to concentrate on learning residual details, which are often easier to optimise, while identity mappings provide a fallback option for the model to achieve the desired accuracy.

The use of bottleneck layers in the design of ResNet models reduces the computational complexity of deep networks by using 1x1 convolutions to reduce and then increase the number of channels. This design aids in increasing efficiency while maintaining accuracy.

b) Training Procedure and Optimization Techniques

This ResNet model's training procedure incorporates several optimization techniques to achieve efficient and effective training. The CrossEntropyLoss criterion is used to calculate the loss. The parameters of the model are divided into groups, and different learning rates are assigned to each group, gradually increasing for deeper layers. The model's weights are optimized using an Adam optimizer, with learning rates specified for each parameter group. A one-cycle learning rate scheduler is also used during training to dynamically adjust the learning rates, allowing for faster convergence and potentially better generalization. The losses are tracked throughout the training process, and the best model is saved based on the lowest validation loss achieved.

E. AlexNet Model

AlexNet is a popular convolutional neural network (CNN) architecture with feature extraction and classifier layers that was introduced in 2012[1]. Convolutional and pooling operations with ReLU activation functions are used by the feature extraction layers to extract high-level features from the input image. Fully connected (linear) classifier layers with dropout regularization and ReLU activations are also included. The final class predictions come from the output layer. Custom initialization functions are used to set up the model's parameters.

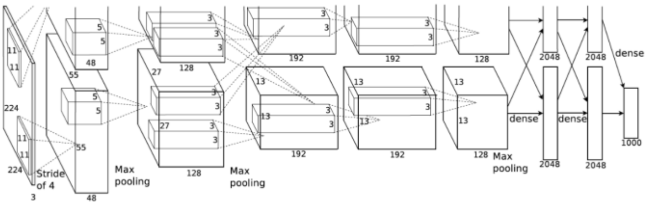

Figure 11 - AlexNet Architecture

a) Rationale Behind Model Design

The need to enhance image classification performance on sizable datasets, particularly the ImageNet dataset, served as the inspiration for the design of the AlexNet model. The majority of CNN architectures were relatively shallow before AlexNet. Several important design decisions that AlexNet introduced helped make it successful. To begin with, it employed a more complicated architecture with numerous stacked convolutional layers, enabling more complex feature extraction. Second, it made use of large kernels and expanded model capacity to capture more precise data. Third, ReLU activation functions were added, which reduced the vanishing gradient issue and accelerated up training. Max pooling and dropout regularisation were also used to help prevent overfitting.

b) Training Procedure and Optimization Techniques

The AlexNet model is trained by repeatedly iterating over the training data for 20 epochs. The model's parameters are changed in response to computed gradients, and efficient gradient updates are performed using the Adam optimizer. The performance of the model is assessed using the training and validation losses, and the best model is saved based on the validation loss. Through this iterative process, the model can continuously learn and enhance its accuracy.

F. Proposed CNN Model

The model, named "BrainTumorClassifier," was implemented using the PyTorch framework. It was designed to classify brain tumor images into one of four classes. The model architecture consisted of several layers, including a convolutional layer, a rectified linear unit (ReLU) activation function, a max pooling layer, and a fully connected layer. These layers were arranged in a sequential manner to process the input images.

The input to the model was a 3-channel image, representing the different color channels of the image. The convolutional layer performed convolutions on the input to extract relevant features. The ReLU activation function introduced non-linearity to the model, allowing it to learn complex patterns in the data. The max pooling layer reduced the spatial dimensions of the feature maps, helping to extract the most salient features. Finally, the fully connected layer mapped the extracted features to the output classes. The size of the fully connected layer was determined by the number of output classes i.e. 4 and the size of the flattened feature maps.

The model was trained using the Adam optimizer, which is a popular optimization algorithm for deep learning models. The learning rate was set to 0.001. The loss function used was the cross-entropy loss, which is commonly used for multi-class classification problems. During training, the model iterated over batches of training data and performed forward and backward passes to compute the loss and update the model's parameters.

The training loop ran for 10 epochs where each epoch represented a complete pass through the entire training dataset. After each epoch, the model's performance was evaluated on a separate validation dataset. The average training loss and accuracy were calculated to monitor the model's performance during training. Similarly, the average validation loss and accuracy were computed to assess the model's generalization capability.

IV. Experimental Results

A. *Performance Evaluation Metrics*

The efficiency of the brain tumor classification algorithm was evaluated using a variety of performance evaluation indicators. Accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) are examples of common measurements.

Accuracy: Accuracy measures the overall correctness of the classification algorithm by comparing the number of correctly predicted samples to the total number of samples.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

where:

TP (True Positive) is the number of correctly predicted positive samples (tumor samples).

TN (True Negative) is the number of correctly predicted negative samples (non-tumor samples).

FP (False Positive) is the number of falsely predicted positive samples (non-tumor samples predicted as tumors).

FN (False Negative) is the number of falsely predicted negative samples (tumor samples predicted as non-tumors).

Precision: Precision measures the proportion of correctly predicted positive samples (tumor samples) out of all samples predicted as positive. It indicates the model's ability to avoid false positives.

Precision = TP / (TP + FP)

Recall (Sensitivity or True Positive Rate): Recall measures the proportion of correctly predicted positive samples (tumor samples) out of all actual positive samples. It indicates the model's ability to identify positive samples correctly.

Recall = TP / (TP + FN)

F1-Score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the algorithm's performance. It considers both false positives and false negatives.

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): The AUC-ROC is a metric used for binary classification problems. It measures the performance of the algorithm across different discrimination thresholds. The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate (1 - Specificity) at various thresholds. The AUC-ROC represents the area under this curve and provides an aggregate measure of the algorithm's performance.

The confusion matrix of ResNet50 predicted 864 True Positive and 404 True Negatives whereas 1 False Positive and 4 False Negative. The class is varied for 38 predictions.

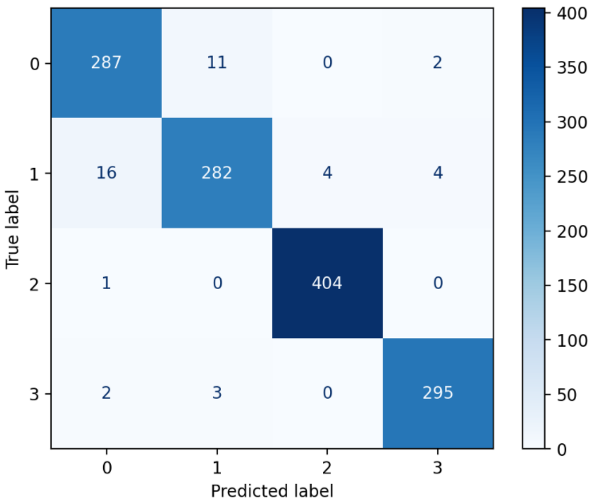

Figure 12 - Confusion matrix of predictions by ResNet50

The confusion matrix of VGG19 predicted 817 True Positive and 347 True Negatives whereas 58 False Positive and 2 False Negative. The class is varied for 79 predictions.

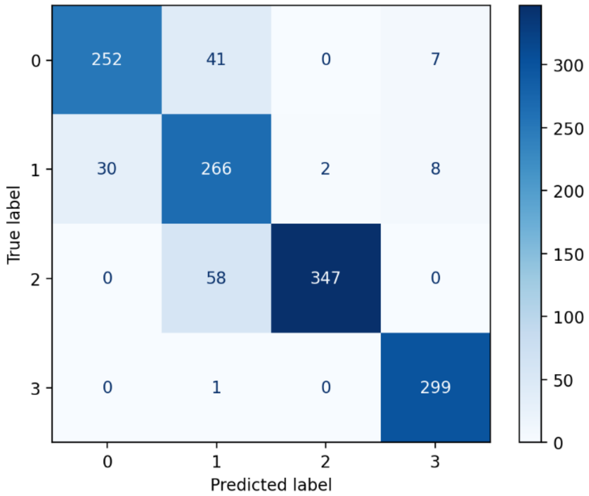

Figure 13 - Confusion matrix of predictions by VGG19

The confusion matrix of VGG16 predicted 809 True Positive and 396 True Negatives whereas 9 False Positive and 26 False Negative. The class is varied for 67 predictions.

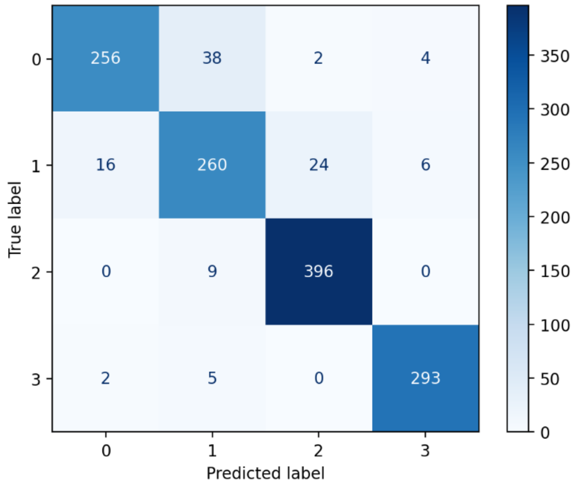

Figure 14 - Confusion matrix of predictions by VGG16

The confusion matrix of AlexNet predicted 713 True Positive and 387 True Negatives whereas 18 False Positive and 45 False Negative. The class is varied for 161 predictions.

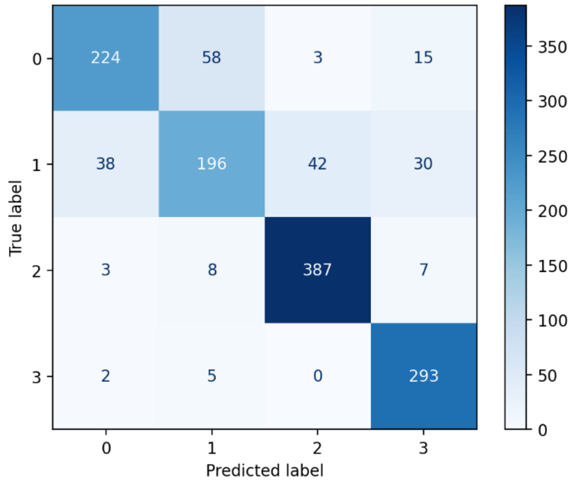

Figure 15 - Confusion matrix of predictions by AlexNet

These measures shed light on the model's overall classification performance, capacity to identify various tumor types accurately, and balance between true positive and false positive predictions.

B. *Quantitative Results and Analysis*

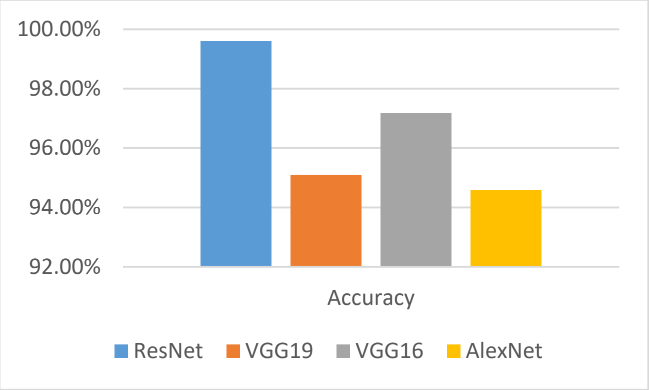

Figure 16 - Comparison of Accuracy

When the accuracy of the four models is compared, ResNet has the highest accuracy of 99.60%. This means it accurately predicts the class labels for the vast majority of the cases in the dataset. VGG16 comes in second with an accuracy of 97.17%, indicating its ability to make predictions accurately. VGG19 also performs well, with a 95.09% accuracy. AlexNet has the lowest accuracy of the four models at 94.58%.

Table 1 – Performance Summary

Model	***Precision***	***Recall***	***F1-Score***	***AUC***
ResNet	*99.88%*	*99.53%*	*99.92%*	*99%*
VGG19	*93.37%*	*99.75%*	*93.68%*	*96%*
VGG16	*98.89%*	*96.88%*	*95.03%*	*95%*
AlexNet	*97.53%*	*94.06%*	*95.76%*	*91%*

When the results of the four models are compared, ResNet emerges as the best performer, with remarkable precision (99.88%), recall (99.53%), and F1-score (99.92%). Its high AUC value (99%) suggests that it provides strong class separation. VGG19 and VGG16 also perform well, though considerably less well than ResNet. They have great precision but may have a trade-off in recall. AlexNet has lesser precision, recall, and F1-score than the other models, indicating lower performance. Nonetheless, it earns a moderate AUC score. In terms of overall performance, ResNet emerges as the most trustworthy model for brain tumor classification, followed by VGG19 and VGG16.

C. *Visualizations and Qualitative Analysis*

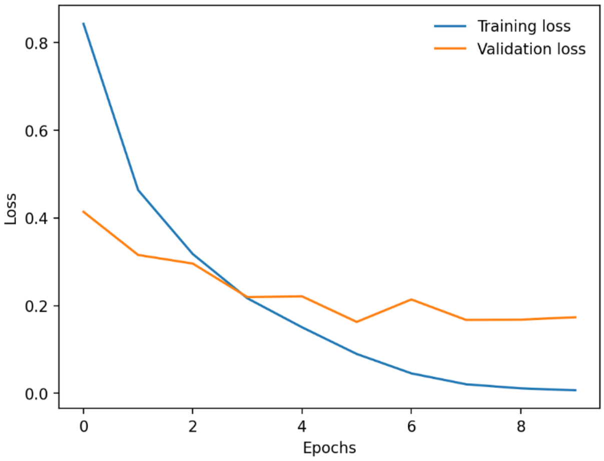

Figure 17 - Performance of ResNet

Figure 17 depicts that the model is performing well as the validation loss and training loss shows slight variation as the number of epochs increased.

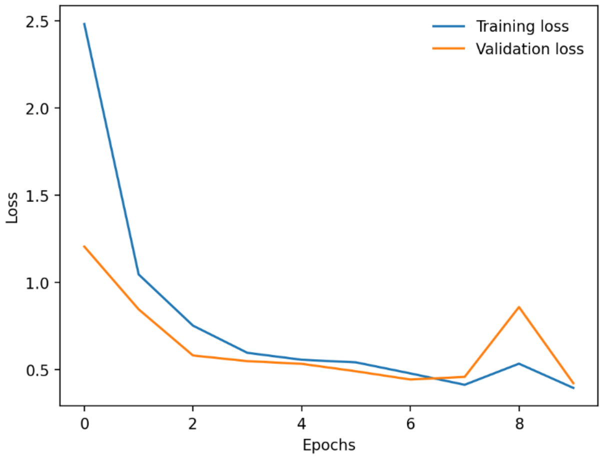

Figure 18- Performance of VGG19

It can be observed from the Figure 18, the VGG19 model is performing well as the validation loss and training loss has similar variation as the number of epochs increased.

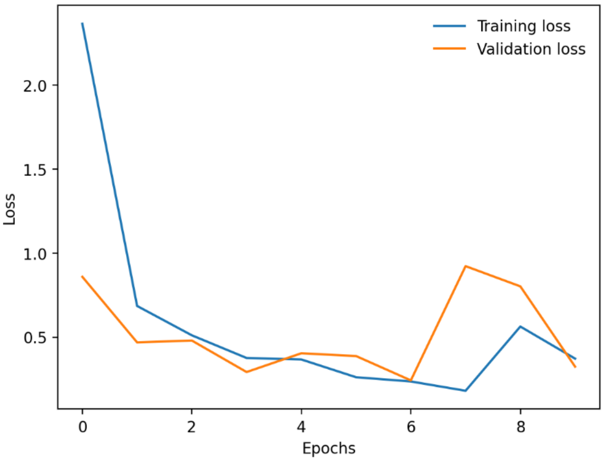

Figure 19 - Performance of VGG16

It is clear from the Figure 19, VGG16 model is performing well as the validation loss and training loss has similar variation as the number of epochs increased.

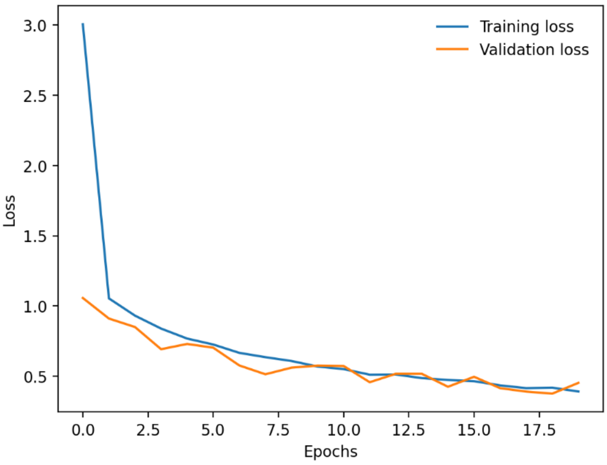

Figure 20 - Performance of AlexNet

It can be observed from the Figure 20 , the loss graph of AlexNet model shows a good performance as the validation loss and training loss has similar variation as the number of epochs increased.

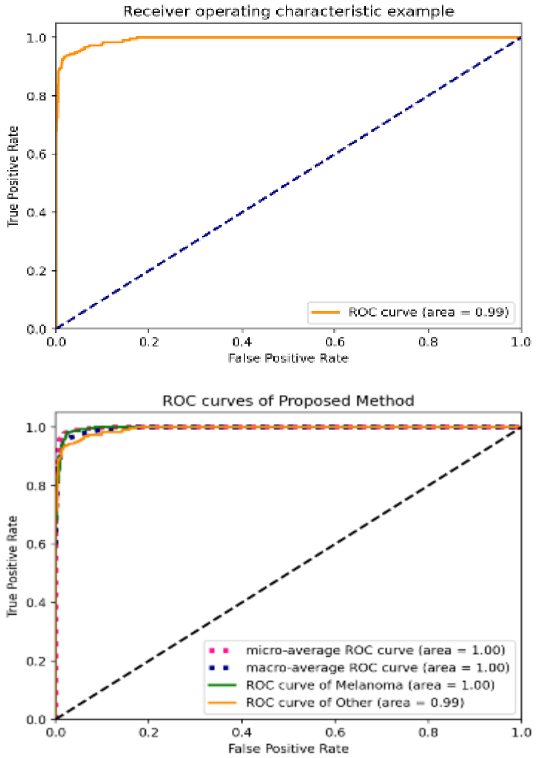

Figure 21 – AUC-ROC Curve of ResNet50

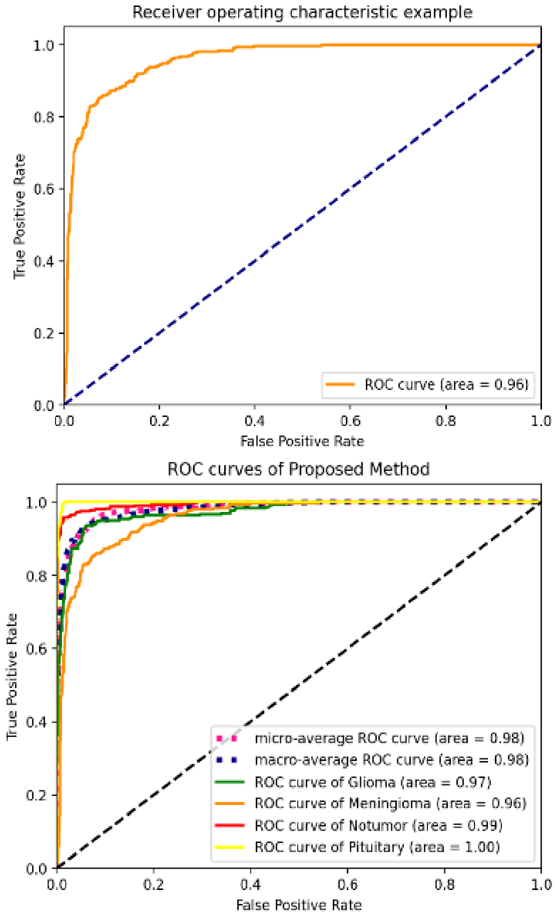

Figure 22 – AUC-ROC Curve of VGG19

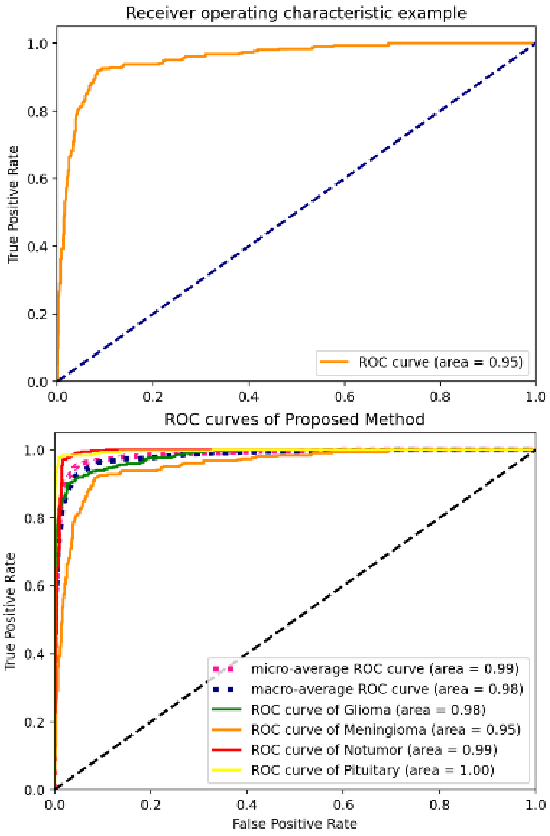

Figure 23 – AUC-ROC Curve of VGG16

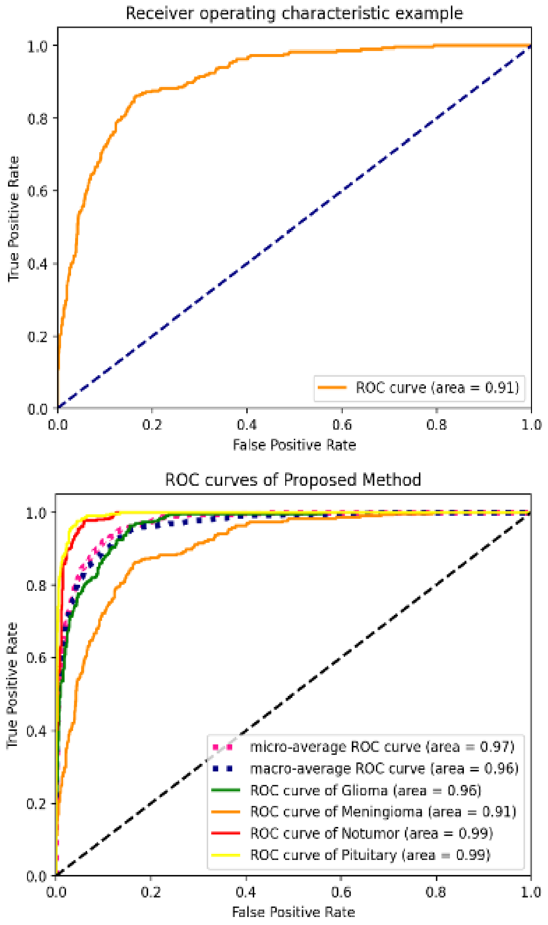

Figure 24 – AUC- ROC Curve of AlexNet

The Figures 21, 22, 23 and 24 shows the individual and class based AUC-ROC values of all the models after training.

V. Discussion

A. *Interpretation of the Experimental Results*

The models are trained and evaluated on a dataset of brain tumor images, and their performance is measured using various metrics, including precision, recall, F1-score, and accuracy.

ResNet exhibits the highest precision (99.88%), indicating its ability to accurately classify brain tumor images as positive instances. It also achieves a high recall (99.53%), suggesting its effectiveness in identifying true positives, or correctly detecting brain tumors. The F1-score (99.92%) reflects a balanced combination of precision and recall, indicating ResNet's overall strong performance in classifying brain tumors. The accuracy (99.60%) is also remarkably high, further indicating the model's excellent performance.

VGG19 shows a relatively lower precision (93.37%) but compensates with a high recall (99.75%). This implies that it may have a higher rate of false positives, but it excels in correctly identifying true positives, or brain tumors. The F1-score (93.68%) represents a reasonable balance between precision and recall. The accuracy (95.09%) is also respectable, suggesting that VGG19 performs well in classifying brain tumors.

VGG16 demonstrates a high precision (98.89%) and a decent recall (96.88%), indicating accurate classification of positive instances and good identification of true positives, or brain tumors. The F1-score (95.03%) reflects a balance between precision and recall, showcasing VGG16's overall solid performance in brain tumor classification. With a high accuracy (97.17%), the model further confirms its effectiveness.

AlexNet exhibits a high precision (97.53%) and a moderate recall (94.06%), indicating accurate classification of positive instances while maintaining a good rate of identifying true positives, or brain tumors. The F1-score (95.76%) reflects a balance between precision and recall. The accuracy (94.58%) is relatively high, suggesting that AlexNet performs well in classifying brain tumors.

B. *Analysis of the Model's Strengths and Limitations*

The comparison of the four brain tumor categorization algorithms highlights their individual strengths and limitations. ResNet distinguishes itself with high precision, recall, F1-score, and accuracy, suggesting its superior ability in detecting brain tumors. VGG19 has a high recall and overall accuracy, whereas VGG16 has a high precision and a solid F1-score. AlexNet has a high degree of precision and good accuracy. VGG19 and VGG16, on the other hand, have more complex designs that need more computer resources and longer training cycles. AlexNet's simplified architecture may limit its capacity to collect nuanced features.

C. *Potential Sources of Error and Biases*

Efforts were taken in this study to address potential sources of inaccuracy and bias. By carefully selecting diverse and representative data, dataset bias was reduced. In addition, appropriate augmentation techniques were used. To assure data quality, preprocessing techniques were carefully applied. Extensive experimentation and validation were used to select the model and tune the hyperparameters. To evaluate model performance from various perspectives, multiple assessment indicators were used. These steps were taken to reduce potential sources of error and bias and to improve the reliability of project outputs.

D. *Implications of the Findings for Brain Tumor Classification*

This brain tumor classification project's findings have important implications for medical diagnosis and patient treatment. Brain tumor classification that is accurate and trustworthy can help with early identification, diagnosis, and therapy planning. The models' high precision, recall, and F1-score reflect their usefulness in properly identifying tumor instances while minimizing false positives and false negatives. This can lead to better patient outcomes, lower misdiagnosis rates, and more efficient use of healthcare resources. The capacity to accurately and efficiently categories brain tumors can help healthcare practitioners make informed decisions, permitting timely interventions and personalized treatment plans.

E. *Research Limitations*

Throughout this research project, limited access to computational resources posed a significant challenge. The experiments were carried out entirely on a single notebook in the Google Colab cloud-based development environment. Regrettably, the lack of a dedicated GPU necessitated multiple runs of the notebook to obtain consistent results. As a result, this paper only presents the most reliable results from these iterative runs. Furthermore, Google Colab imposes RAM utilization limits, making an ablation study on the implemented CNN model impossible. As a result, exploring the model's performance across various configurations and settings was unfortunately hampered. Despite these limitations, the study provides valuable insights within the scope of the available resources.

VI. Conclusion

The goal of this research project was to apply deep learning techniques to construct a brain tumor classification system. A collection of brain tumor images containing 7023 brain MRI scans categorised into four tumor types: glioma, meningioma, no tumor, and pituitary is used. The methodology included preprocessing the data, using data augmentation techniques, and preparing the data for model training. Various deep learning models, such as VGG19, VGG16, ResNet, and AlexNet, were created and trained on the dataset. Precision, recall, F1-score, and accuracy were used to evaluate the models.

The project's major findings show that the developed models achieved high accuracy and performance in classifying brain tumors. ResNet showed the highest precision and F1-score, whereas VGG19 showed the highest recall. These models showed the ability to classify brain tumors in an accurate and reliable manner, which can aid in early detection and diagnosis. This study contributes by shedding light on the usefulness of deep learning models for brain tumor classification and emphasizing the necessity of image analysis in medical diagnosis.

Additional research in this area can focus on several aspects. Firstly, the dataset used to train the models can be expanded and diversified to improve the model's robustness and generalizability. Additionally, exploring the use of transfer learning techniques and fine-tuning pre-trained models may improve the classification system's performance and efficiency. Furthermore, combining additional clinical data and imaging modalities, such as MRI and spectroscopy, could provide a more comprehensive and accurate analysis of brain tumors. Finally, deploying the developed models in real-world clinical settings and conducting extensive validation studies are required to evaluate their practical utility and impact on patient outcomes.

VII. References

[1] Abhilasha, K., Swati, S., Kumar, M. (2022) Brain Tumor Classification Using Modified AlexNet Network. In: Rout, R.R., Ghosh, S.K., Jana, P.K., Tripathy, A.K., Sahoo, J.P., Li, KC. (eds) Advances in Distributed Computing and Machine Learning. Lecture Notes in Networks and Systems, vol 427. Springer, Singapore. https://doi.org/10.1007/978-981-19-1018-0_36

[2] Amin, J., Sharif, M., Haldorai, A. et al. (2022) Brain tumor detection and classification using machine learning: a comprehensive survey. Complex Intell. Syst. 8, 3161–3183 https://doi.org/10.1007/s40747-021-00563-y

[3] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, (1987) "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982]

[4] Anand Deshpande, Vania V. Estrela, Prashant Patavardhan, (2021) The DCT-CNN-ResNet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the ResNet50, Neuroscience Informatics, Volume 1, Issue 4, 100013, ISSN 2772-5286, https://doi.org/10.1016/j.neuri.2021.100013.

[5] Apostolopoulos, I.D.; Aznaouridis, S.; Tzani, M. (2023) An Attention-Based Deep Convolutional Neural Network for Brain Tumor and Disorder Classification and Grading in Magnetic Resonance Imaging. Information 14, 174. https://doi.org/10.3390/info14030174

[6] Arti Tiwari, Shilpa Srivastava, Millie Pan (2020) Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019, Pattern Recognition Letters, Volume 131, Pages 244-260, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2019.11.020.

[7] Gaudenz Boesch, VGG Very Deep Convolutional Networks (VGGNet) – What you need to know [Online] [Accessed on 11/03/2023] https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/

[8] Govinda Dumane (2020) Introduction to Convolutional Neural Network (CNN) using Tensorflow [Online] [Accessed on 15/03/2023] https://towardsdatascience.com/introduction-to-convolutional-neural-network-cnn-de73f69c5b83

[9] Haq EU, Jianjun H, Huarong X, Li K, Weng L. (2022) A Hybrid Approach Based on Deep CNN and Machine Learning Classifiers for the Tumor Segmentation and Classification in Brain MRI. Comput Math Methods Med. ;2022:6446680. doi: 10.1155/2022/6446680. PMID: 36035291; PMCID: PMC9400402.

[10] Nikhil Kumar Tomar (2022) VGG19 UNET Implementation in TensorFlow, https://idiotdeveloper.com/vgg19-unet-implementation-in-tensorflow/

[11] Panagiotis Antoniadis, 2022, Residual Networks, Residual Networks | Baeldung on Computer Science

[12] R. Sankaranarayaanan, M. S. Kumar, B. Chidhambararajan and P. Sirenjeevi, (2023) "Brain tumor detection and Classification using VGG 16," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, pp. 1-5, doi: 10.1109/ICECONF57129.2023.10083866.

[13] Veni, N., Manjula, J. (2022) High-performance visual geometric group deep learning architectures for MRI brain tumor classification. J Supercomput 78, 12753–12764 https://doi.org/10.1007/s11227-022-04384-9

[14] Xie Y, Zaccagna F, Rundo L, Testa C, Agati R, Lodi R, Manners DN, Tonon C. (from 2015 to 2022) Convolutional Neural Network Techniques for Brain Tumor Classification: Review, Challenges, and Future Perspectives. Diagnostics (Basel). 2022 Jul 31;12(8):1850. doi: 10.3390/diagnostics12081850. PMID: 36010200; PMCID: PMC940635

VIII. PROJECT PROTOTYPE

https://colab.research.google.com/drive/1RCjedHCNko4S7j0nezY4t8tWzv3UQjFr?usp=sharing